Cromemco Software Update Service Note C-2

Date: January 4, 1983

Product: CCC-L and CCC-S

Version.Release: 2

Date production of this version began: January 4, 1983

First serial number with this version: 2-10000 on 8"
2-10000 on 5"

SUMMARY

Version 05.10 of the Cromemco C Compiler is now available and updates the previous release, version 05.00. Version 05.10 is compatible with version 11.11 or higher of the Cromix Operating System, and with version 2.52 or higher of CDOS. This version contains fixes for several previously existing bugs. Version 05.10 is enhanced with a new double-precision math library which includes trigonometric and natural logarithmic functions. This release of C provides added system call functions to implement Cromix system calls, including Kill, Pipe and Signal. Several existing system call functions have been changed in their calling sequence, paralleling changes made to the Cromix Operating System. Cromemco's assembly language debugger, Debug, has been added to the distribution package of the C Compiler.

ADDITION OF DEBUG.COM

The assembly language debugger Debug, version 00.17 has been added to the C compiler distribution diskette. Refer to the **Cromemco Debug Instruction Manual** dated June 1981, part number 023-4038, for more information.

MODIFICATION TO MISCELLANEOUS FUNCTIONS

System (cmd)
It is no longer necessary to begin the command string with "shell -c". **System** will now cause the invoking process to ignore the Sigabort (**CONTROL-C**) and Sigterm signals while the child process is executing. This prevents the possibility of killing a parent process but not the child process

in the case where the child process ignores the Sigabort or Sigterm signals. Such a situation could result in a child process which, when finished, would have nowhere to return.

MODIFICATIONS TO SYSTEM CALLS

The following system calls have changed in the calling sequence corresponding to changes made in the Cromix Operating System:

chkdev Checks for presence of device driver

called by:
```
chkdev (dtype, dmajor, dminor)
int dtype
int dmajor
int dminor
```

returns:
```
int 0

or

-1 if error, and
int value from Cromix in errno
```

cstat Determines status of file which is open

called by:
```
cstat (channel, statustype, inodebuffer)
int channel
int statustype
char inodebuffer[128]
```

returns:
```
char inodebuffer[128]

or one of:

int owner, group, owneraccess, groupaccess,
otheraccess nlinks, inodenumber, file type

or

int major_and_minor_devicenumber

or

long filesize

or one of:

char tcreated[6], tmodified[6], taccessed[6],
tdumped[6]
```

or

-1 if error, and
int value from Cromix in errno

fexec Forks and executes a program

called by: fexec (pathname, argv, sigmask, sigvalues)
char *pathname
char *argv[]
int sigmask
int sigvalues

returns: int childprocessnumber

or

-1 if error, and
int value from Cromix in errno

fshell Forks a Shell process

called by: fshell (argv, sigmask, sigvalues)
char *argv
int sigmask
int sigvalues

returns: int processid

or

-1 if error, and
int value from Cromix in errno

fstat Determines status of a file

called by: fstat (pathname, statustype, inodebuffer)
char *pathname
int statustype
char inodebuffer[128]

returns: char inodebuffer[128]

or one of:

int owner, group, owneraccess, groupaccess,
otheraccess nlinks, inodenumber, file type

or

int major_and_minor_devicenumber

or

long filesize

or one of:

char tcreated[6], tmodified[6], taccessed[6], tdumped[6]

or

-1 if error, and
int value from Cromix in errno

makdev Creates a new name for a device

called by: makdev (pathname, dtype, dmajor, dminor)
char *pathname
int dtype
int dmajor
int dminor

returns: int 0

or

-1 if error, and int error code in errno

NEW SYSTEM CALLS

The following functions which implement Cromix system calls have been added to the C library:

alarm Sends an alarm signal to the current process after the specified number of seconds

called by: alarm (snum)
int snum

returns: the priority number [-40..+40] -40 is highest

exchg Exchanges the contents of two inodes

called by:
```
exchg (ichannel, ochannel)
int ichannel
int ochannel
```

returns: int 0

or

-1 if error, and int error code in errno.

getprior Gets the priority number of a current process

called by:
```
getprior ()
```

returns: the priority number [-40..+40] -40 is highest

kill Sends a signal to a process

called by:
```
kill (pid,stype)
int pid
int stype
```

returns: int 0

or

-1 if error, and int error code in errno

lock Makes (or attempts to make) a lock table entry

called by:
```
lock (lock_sequence, ltype, llength)
char *lock_sequence
int ltype
int llength
```

returns: int 0

or

-1 if error, and int error code in errno

pause Sets up to wait for a SIGALARM signal

called by: pause ()

returns: nothing

pipe Creates a pipe

called by: pipe (&pipeout, &pipein)
int pipein
int pipeout

returns: int 0

and

channel number of pipe's receiving end in pipein

channel number of pipe's sending end in pipeout

or

-1 if error, and int error code in errno

setprior Sets the priority number of the current process

called by: setprior (pnum)
int pnum

returns: the priority number [-40..+40] -40 is highest

signal Sets up to receive a signal

called by: signal (execution_address, stype)
char *execution_address
int stype

returns: nothing

sleep Sleeps for a specified number of seconds

called by: sleep (numseconds)
int numseconds

returns: nothing

unlock Removes a lock table entry

called by:
```
unlock (lock_sequence, ltype, llength)
char *lock_sequence
int ltype
int llength
```

returns: nothing

NEW MATH LIBRARY

Transcendental mathematical functions including natural logarithms and trigonometric functions are now available in the C library. Most functions have been tested to return results accurate to between 6 and 12 significant digits.

The trigonometric functions are capable of operating in either **degree** or **radian** mode; the mode must be set prior to calling the trigonometric function. **Degree** or **radian** mode is set by calling either

degrees()
or
radians().

A mode remains in effect unless it is specifically changed by setting the other mode.

Be sure to declare the math functions external and double before calling them.

ln(x)
double x

Returns double: log base e of x

exp (x)
double x

Returns double: e to the power of x

sqrt (x)
double x

Returns double: square root of x

dpwrd (x1, x2)
double x1
double x2

Returns double: x1 to the power of x2

sgn (x)
double x

Returns double: +1L if x is positive
0L if x is zero
-1L if x is negative

abs (x)
double x

Returns double: the absolute value of x

truncate (x)
double x

Returns double: the integer part of x

fract (x)
double x

Returns double: the fractional part of x

degrees()
sets degree mode for trigonometric functions

radians()
sets radian mode for trigonometric functions

sin (x)
double x

Returns double: sine of x

cos (x)
double x

Returns double: cosine of x

tan (x)
double x

Returns double: tangent of x

cot (x)
double x

Returns double: cotangent of x

sec (x)
double x

Returns double: secant of x

csc (x)
double x

Returns double: cosecant of x

asin (x)
double x

Returns double: arcsin of x

acos (x)
double x

Returns double: arccosine of x

atn (x)
double x

Returns double: arctangent of x

acot (x)
double x

Returns double: arccotangent of x

sinh (x)
double x

Returns double: hyperbolic sine of x

cosh (x)
double x

Returns double: hyperbolic cosine of x

tanh (x)
double x

Returns double: hyperbolic tangent of x

coth (x)
double x

Returns double: hyperbolic cotangent of x

sech (x)
double x

Returns double: hyperbolic secant of x

csch (x)
double x

Returns double: hyperbolic cosecant of x

asinh (x)
double x

Returns double: inverse hyperbolic sine of x

cosh (x)
double x

Returns double: inverse hyperbolic cosine of x

acoth (x)
double x

Returns double: inverse hyperbolic cotangent of x

asech (x)
double x

Returns double: inverse hyperbolic secant of x

acsch (x)
double x

Returns double: inverse hyperbolic cosecant of x

New Global Variables in the Library

Both are type **char:**

_overflow 0 if no overflow on previous floating operation
1 if overflow occurred

__degree 0 if arguments and function values are in radians
1 if arguments and function values are in degrees

Number Conversion Routines

double fkey(float, *char)
double dkey(double, *char)

Convert a float or double to a 4 or 8 byte char string respectively, such that the string is sortable by simple Ascii-like comparisons; the converted number is returned to C as the value of the function.

NOTE: if *char is null, the string is not stored.

keyf(*char)
keyd(*char)

Convert a 4 or 8 byte char string, respectively, to a float or double; presumes that the string was created by fkey or dkey, respectively.

int ikey(int, *char)
unsigned ukey(unsigned, *char)

Convert an int or an unsigned, respectively, to 'sortable' order; put result into string pointed to by *char; also return converted int or unsigned, respectively, as value of the function.

NOTE: if *char is null, string store is not done.

keyi(*char)
keyu(*char)

Convert a 2-byte character string to int or unsigned, respectively. Presumes the string was created by ikey or ukey, respectively.

CHANGES TO HEADER FILES

The header files are located in the **/usr/include** directory. Many of these files were modified. Complete listings of the modified files are provided and may replace their old listings in the C manual. Descriptions of changes are:

Structs.h

This is a new header file which defines the structures of the super block and inodes. Note that the bytes composing long, integer, and unsigned values must be reversed. This is because Cromix stores the numerical values msb-to-lsb; C expects the values lsb-to-msb.

Jsysequ.h

This header file changed according to capabilities added to make new Cromix system calls available from C. This file corresponds with the Cromix **jsysequ.z80** file.

Included in the changes are:

New equates for system call numbers are: _exchg, _pipe, _getprior, _setprior, _lock, _unlock, _signal, _kill, _sleep, _alarm, _pause.

A file type of is_pipe has been added.

New signal types are: sigabort, siguser, sigkill, sigterm, sigalarm, sigpipe, sighangup, sigmax.

Several new error codes have been added.

Modeequ.h

The **modeequ.h** file has changed considerably between this release of C and the previous release of C version 5.00. C version 5.00 was compatible with Cromix version 10.09, while C version 5.10 is compatible with Cromix version 11.11 and higher.

The new **modeequ.h** file corresponds to changes made to procedures of the Cromix **.getmode** and **.setmode** system calls. Many of the defines underwent name changes, with significant old defines remaining effective by being redefined to their new names.

The following defines are no longer used by the Cromix Operating System and have been eliminated from the **modeequ.h** file:

md_iwake	md1v_tty
md_owake	md2v_tty
md_nlnulls	md1v_outp
md_tabnulls	md2v_outp
md_ffnulls	md1v_inp
md_crnulls	md2v_inp
b_50	st_linerdy
b_75	st_keybd
b_134	st_signal
b_200	st_abort
b_600	id_tty
b_1800	id_output
b_wakeup	id_serial
b_noway	id_nochg
md_outraw	

For more information concerning the parameters of the **.getmode** and **.setmode** system calls, refer to **Cromemco Cromix Operating System Instruction Manual** dated June 1982, part number 023-4022.

Cdoscalls.h

This header file changed according to additional or modified system calls implemented in **sim.bin**, Cromix's CDOS simulator. The new call-definitions along with their CDOS system call numbers are: _cdconsio (6), _cpversion (12), _cdwrprot (28), _cdgetrovect (29), _cdsetattr (30)_, _cdusrcode (32), _cdrdrand (33), _cdwrrand (34), _cdfsize (35), _cdsetrand (36), _cdwrzrand)40_, _cdrdtrack (160), _cdwrtrack (161). Previously _cddeselect was assigned to be system call (12); it is now system call (162).

For detailed information on the individual system calls, refer to the **CDOS-1** suds note dated July 15, 1982, part number 023-9533.

CORRECTIONS TO THE MANUAL

The following corrections apply to the **C Programming Language Instruction Manual** dated February 1981, part number 023-4029.

The descriptions of the results returned from the functions **fscanf**, **scanf** and **sscanf** in Chapter 4, Input and Output, should read as follows:

Returns **int**:

number of fields scanned;
-1 if EOF encountered.

The description for the **delete** function in Chapter 6, System Calls, should read as follows:

delete removes one directory entry. If the directory entry is the last entry for that file then the operating system will remove the file from the file system.

BUGS THAT HAVE BEEN FIXED

Pre-incrementing and post-incrementing and pre-decrementing of float variables did not work. This has been fixed by correcting the macros in the **cmacros.Z80** file which implement these functions.

Previously **fscanf** erroneously returned a 0 on EOF. This has been fixed and a -1 is returned.

Previously if **scanf** read a float value with an exponent greater than 62, causing overflow, scanf could not read the next field correctly. This has been fixed.

Previously, **open** did not return a -1 when an attempt was made to open a nonexistent file. This has been fixed so an error of -1 is returned.

Previously **getl** erroneously returned a -1 on EOF. This has been fixed and a 0 is returned.

Previously, the **b** option of the CDOS version of **fopen** did not function correctly. This has been fixed.

Previously **getline** was erroneously defined in the **stdio.h** file to use the **stdin** file structure as its input file argument. It has been changed to use the **STDIN** filepointer.

KNOWN BUGS

The second **float** argument passed to a function is improperly received as 0 because the function expects **double** values as arguments. A user fix is to declare parameters **double** in the called routine.

Expressions nesting **toupper** or **tolower** and **isalpha**, such as:

```
x = isalpha (c = toupper(c))
```

cause the compiler to loop because both are defined in terms of **islower** or **isupper** in **stdio.h** and **cdstdio.h**. A user fix is to either separate the expression into two expressions, or redefine one of the functions to not use **islower** or **isupper**. For example, **isalpha** may be redefined as follows:

```
#define isalpha(c) (( 'A'<=(c)&&(c)<='Z' )||( 'a'<=(c)&&(c)<='z'))
```

Casts of **float** or **double** to **integer** or **long** do not work. Type conversion will, however, work on variable assignments, so one can assign an **integer** variable to a **float** variable.

A relational expression in an **if** statement, which uses the result returned from a non-integer function may not work properly. A fix is to evaluate the function to an intermediary variable and then use the variable in the relational expression.

In an assignment statement of the form

```
X = function1()  +  expression2
```

where function1 is a function which returns **float** or **long**, the result of function1 is unpredictable if expression2 is

a. of the form x1 op x2, where op is +, -, *, / or %
b. another function
c. a value designated by a pointer variable

A user fix is to evaluate function1 through an intermediary variable.

C does not check for stack overflow until the first function is called. If stack overflow occurs before a function call, the program will run away and be aborted by the operating system.

Putc does not return -1 on failure to write to a file opened for read access.

A character variable is not properly assigned the value designated by **\xhh** where h is a hexadecimal digit.

The CDOS **scanf** does not echo a line feed after <CR> and last line entered is overwritten.

Trying to **fclose** a file with a null file pointer causes **fatal file close error** even when no **e** option was used in **fopen.**

The **unlink** function is presently implemented as the **delete** function. **Unlink** is currently not available under CDOS.

Initialization of an external or static structure at declaration time fails for fields of the structure which follow a string-initialized field of the structure.

Using the **?:** operator within a **printf** statement to determine a **printf** parameter causes output of garbage or a runaway program if the **printf** statement also has parameters of type long, float or double. For example, the following will fail:

```
int i;
double *p;
printf("%lf%c", *p++, (i==0 ?  '\n':  ' ') );
```

The compiler will not accept taking the **sizeof** a variable designated by a pointer. For instance, the compiler will not accept:

```
int i, *j;
i=sizeof (*j);
```

STANDARD HEADER FILES

cdoscalls.h

```
#control nlist

/* cdoscalls.h:  Cromemco C I/O header file

  Copyright (c) 1980 by Cromemco, Inc., All Rights Reserved

  This file contains definitions for all CDOS system calls
  which can be made using the functions rcdos and ccdos

  Compatible with CDOS version 2.52 or higher and
  Cromix version 11.11 or higher
*/


#define _cdabort          0
#define _cdrdcons         1
#define _cdwrcons         2
#define _cdrdrdr          3
#define _cdwrpun          4
#define _cdwrlpt          5
#define _cdconsio         6
#define _cdgetiob         7
#define _cdsetiob         8
#define _cdputl           9
#define _cdgetl           10
#define _cdtestcons       11
#define _cpversion        12
#define _cdboot           13
#define _cdselect         14
#define _cdopen           15
#define _cdclose          16
#define _cdsearch         17
#define _cdfindnext       18
#define _cddelete         19
#define _cdreadnext       20
#define _cdwrnext         21
#define _cdcreate         22
#define _cdrename         23
#define _cdgetlogin       24
#define _cdcurrent        25
#define _cdsetbuff        26
#define _cdgetamap        27
#define _cdwrprot         28    /* ignored in CDOS and Sim */
#define _cdgetrovect      29    /* ignored in CDOS and Sim */
#define _cdsetattr        30    /* ignored in CDOS and Sim */
/*                        31       illegal in CDOS and Sim */
#define _cdusrcode        32
```

```
#define _cdrdrand       33
#define _cdwrrand       34
#define _cdfsize        35
#define _cdsetrand      36
/*                      37      ignored in CDOS and Sim */

#define _cdwrzrand      40

#define _cdrdnoecho     128
#define _cdgetuser      129
#define _cdsetcc        130
#define _cdrdlogical    131
#define _cdwrlogical    132
#define _cdformfcb      134   /* note:  address of terminator
                                        not returned */
#define _cdupdate       135
#define _cdlink         136
#define _cdmul          137
#define _cddiv          138
#define _cdhome         139
#define _cdeject        140
#define _cdversion      141
#define _cdcrt          142
#define _cdsetdate      143
#define _cdgetdate      144
#define _cdsettime      145
#define _cdgettime      146
#define _cdretcode      147
#define _cdattr         148
#define _cdbottom       151

#define _cdrdtrack      160   /* implemented in CDOS; illegal in Sim */
#define _cdwrtrack      161   /* implemented in CDOS; illegal in Sim */
#define _cddeselect     162

#control list
```

jsysequ.h

```
#control nlist

/* jsysequ.h:  Cromemco C I/O header file

   Copyright (c) 1980 by Cromemco, Inc., All Rights Reserved

   This file contains declarations of all values which are
   used during calls to the Cromix Operating System.

   Compatible with Cromix version 11.05 or higher
*/

/*
   access modes for create
*/

#define op_read         0       /* read only */
#define op_write        1       /* write only */
#define op_rdwr         2       /* read and write */
#define op_append       3       /* append only */
#define op_xread        4       /* exclusive read only */
#define op_xwrite       5       /* exclusive write only */
#define op_xrdwr        6       /* exclusive read and write */
#define op_xappend      7       /* exclusive append only */

#define op_truncf       0x80    /* truncate on create flag */
#define op_condf        0x40    /* conditional create flag */

/*      Modes for setpos system call    */
#define fwd_begin       0       /* Forward from the beginning of the file */
#define fwd_current     1       /* Forward from the current position */
#define fwd_end         2       /* Forward from the end of the file */
#define bak_current     -1      /* Backward from the current file position */
#define bak_end         -2      /* Backward from the end of the file */
/*
   status types for _fstat, _cstat, _fchstat, _cchstat
*/

#define st_all          0       /* all of inode (128 bytes) */
#define st_owner        1       /* owner */
#define st_group        2       /* group */
#define st_aowner       3       /* owner access, mask */
#define st_agroup       4       /* group access, mask */
#define st_aother       5       /* other access, mask */
#define st_ftype        6       /* file type, special device # */
#define st_size         7       /* file size */
#define st_nlinks       8       /* number of links */
#define st_inum         9       /* inode number */
#define st_device       10      /* device containing inode */
#define st_tcreate      11      /* time created */
#define st_tmodify      12      /* time last modified */
```

```
#define st_taccess      13      /* time last accessed */
#define st_tdumped      14      /* time last dumped */
#define st_devno        15      /* device number if inode is a device */

/*
    file types for st_ftype
*/

#define is_ordin        0       /* ordinary file */
#define is_direct       1       /* directory file */
#define is_char         2       /* character device */
#define is_block        3       /* block device */
#define is_pipe         4       /* Pipe */

/*
    mask values for file access flags
*/

#define ac_read         0x01    /* read access bit */
#define ac_exec         0x02    /* execute access bit */
#define ac_writ         0x04    /* write access bit */
#define ac_apnd         0x08    /* append access bit */

/*
   id types and values for _setuser, _getuser, _setgroup, _getgroup
*/

#define id_effective    0       /* effective id */
#define id_login        1       /* login id */
#define id_program      2       /* program id */
#define id_hl           3       /* id contained in idnumber */

/*
   signal types
*/

#define sigabort        1       /* Control-C key */
#define siguser         2       /* User specifiable key */
#define sigkill         3       /* Kill signal */
#define sigterm         4       /* Terminate */
#define sigalarm        5       /* Alarm */
#define sigpipe         6       /* Broken pipe signal */
#define sighangup       7       /* Modem hang up */
#define sigmax          8       /* Maximum signal number */

/*
    Cromix System Call Numbers
*/

#define _makdev         0x00    /* make device entry */
#define _makdir         0x01    /* make a directory */
#define _getdir         0x02    /* get current directory name */
#define _setdir         0x03    /* change current directory */
```

```
#define _mount          0x04    /* mount file system */
#define _unmount        0x05    /* unmount file system */
#define _delete         0x06    /* delete file */
#define _chkdev         0x07    /* check for device driver */

#define _create         0x08    /* create & open file */
#define _open           0x09    /* open file */
#define _chdup          0x0A    /* duplicate channel */
#define _close          0x0B    /* close file */
#define _exchg          0x0C    /* Exchange the contents of two inodes */

#define _trunc          0x0D    /* truncate open file */
#define _pipe           0x0E    /* generate a pipe */

#define _getpos         0x10    /* get file position */
#define _setpos         0x11    /* set file position */
#define _getmode        0x12    /* get device characteristics */
#define _setmode        0x13    /* set device characteristics */

#define _rdseq          0x14    /* read n bytes */
#define _wrseq          0x15    /* write n bytes */
#define _rdbyte         0x16    /* read 1 byte */
#define _wrbyte         0x17    /* write 1 byte */
#define _rdline         0x18    /* read a line */
#define _wrline         0x19    /* write a line */
#define _printf         0x1B    /* print formatted string */
#define _error          0x1C    /* print error message */

#define _fstat          0x20    /* get file status (inode) */
#define _cstat          0x21    /* get channel status (inode) */
#define _fchstat        0x22    /* change file status */
#define _cchstat        0x23    /* change channel status */
#define _flink          0x24    /* link to file */
#define _clink          0x25    /* link to open channel */
#define _faccess        0x26    /* test file access */
#define _caccess        0x27    /* test channel access */

#define _getdate        0x30    /* get date */
#define _setdate        0x31    /* set date */
#define _gettime        0x32    /* get time */
#define _settime        0x33    /* set time */

#define _getuser        0x34    /* get user id */
#define _setuser        0x35    /* set user id */
#define _getgroup       0x36    /* get group id */
#define _setgroup       0x37    /* set group id */
#define _getprior       0x38    /* get the current processes priority number
#define _setprior       0x39    /* set the current processes priority number

#define _getproc        0x3A    /* get process id */
#define _ksam           0x3D    /* ksam call */
#define _lock           0x3E    /* lock key */
#define _unlock         0x3F    /* unlock key */
#define _signal         0x40    /* set up to receive a signal */
```

```
#define _kill           0x41    /* send a signal */
#define _sleep          0x42    /* sleep for specified number of seconds */
#define _alarm          0x43    /* set alarm clock */
#define _pause          0x44    /* pause for alarm clock */

#define _wait           0x45    /* wait for child process */
#define _exit           0x46    /* exit process (close files) */

#define _fshell         0x48    /* fork a shell process */
#define _shell          0x49    /* transfer to shell process */
#define _fexec          0x4B    /* fork and execute program */
#define _exec           0x4C    /* execute program */

/*                      0x50     */
#define _indirect       0x51    /* system call in A-register */

#define _update         0x52    /* update disk I/O buffers */
#define _mult           0x53    /* multiply */
#define _divd           0x54    /* divide */
#define _version        0x55    /* get system version # */
#define _boot           0x56    /* boot operating system */

/*
        Cromix error numbers returned in extern int errno
*/

#define _badchan        1       /* bad channel # */
#define _toomany        2       /* channel already open */
#define _notopen        3       /* channel not open */
#define _endfile        4       /* end-of-file */
#define _ioerror        5       /* I/O error */
#define _filtable       6       /* file table exhausted */
#define _notexist       7       /* file does not exist */
#define _badname        8       /* bad file name */
#define _diraccess      9       /* directory access */
#define _filaccess      10      /* file access */
#define _exists         11      /* file already exists */
#define _nospace        12      /* no disk space left */
#define _noinode        13      /* no inodes left */
#define _inotable       14      /* inode table exhausted */
#define _badcall        15      /* illegal system call */
#define _filsize        16      /* file size too big */
#define _mnttable       17      /* mount table exhausted */
#define _notdir         18      /* not a directory */
#define _isdir          19      /* is a directory */
#define _priv           20      /* privileged system call */
#define _notblk         21      /* not a block special device */
#define _fsbusy         22      /* file system busy */
#define _notordin       23      /* not an ordinary file */
#define _notmount       24      /* device not mounted */
#define _nochild        25      /* no child processes */
#define _nomemory       26      /* not enough memory */
#define _ovflo          27      /* divide overflow */
#define _argtable       28      /* argument table exhausted */
```

```
#define _arglist        29      /* bad argument list */
#define _numlinks       30      /* too many number of links */
#define _difdev         31      /* cross-device link */
#define _nodevice       32      /* no special device */
#define _usrtable       33      /* user process table exhausted */
#define _badvalue       34      /* value out of range */
#define _notconn        35      /* I/O device not connected */
#define _devopen        36      /* device open error                */
#define _diruse         37      /* directory in use (delete) */
#define _filuse         38      /* file in use (exclusive access) */
#define _nomatch        39      /* no match on ambiguous name */
#define _chnaccess      40      /* channel access */
#define _notcromix      41      /* not a cromix disk */
#define _badfree        42      /* bad free list */
#define _badinum        43      /* bad inode number */
#define _readonly       44      /* device mounted for read only */
#define _noproc         45      /* process does not exist */
#define _signal         46      /* system call was aborted */
#define _badpipe        47      /* bad call on pipe */
#define _locked         48      /* locked */
#define _deadlock       49      /* deadlocked */
#define _lcktable       50      /* lock table exhausted */

/* ------------------------------------------------------------------------

/*                      Old Names

#define pos_begin       fwd_begin
#define pos_current     fwd_current
#define pos_end         fwd_end


#control list
```

modeequ.h

```
#control nlist

/*      modeequ.h:    Cromemco C I/O header file

        Copyright (c) 1980 by Cromemco, Inc. All Rights Reserved

        This file contains declarations of all values which are
        used in the getmode and setmode Cromix system calls.

        Compatible with Cromix version 11.08 or higher
*/


#define MD_ISPEED       0               /* input speed */
#define MD_OSPEED       1               /* output speed */
#define MD_MODE1        2               /* flags: RAW, ECHO, etc. */
#define MD_MODED        3               /* delays for NL, CR, etc. */
#define MD_MODE2        4               /* flags: PAUSE, XFF, etc. */
#define MD_MODE3        5               /* flags: ESCRETN */
#define MD_ERASE        6               /* auxiliary erase character */
#define MD_DELECHO      7               /* erasure echo character */
#define MD_LKILL        8               /* line kill character */
#define MD_USIGNAL      9               /* user signal key */
#define MD_LENGTH       10              /* page length (lines) */
#define MD_WIDTH        11              /* page width (columns) */
#define MD_BMARGIN      12              /* bottom margin (lines) */
#define MODELEN         MD_BMARGIN + 1

/* the following are commands, not displacements in the device structure */

#define MD_STATUS       -100            /* flag: character is in one
                                                 of the  input queues */
#define MD_IFLUSH       -101            /* flush input queues */
#define MD_FNKEYS       -104            /* turn function keys on or off */
#define MD_PSIGHUP      -105            /* signal current process if hang u
/*                      -106               (this value reserved) */

/* contents of d-register for MD_ISPEED calls to change the baudrate
   stored in md_ispeed  */

#define S_HANGUP        0       /* hang up dataphone */
/*                      1       ;  50 baud */
/*                      2       ;  75 baud */
#define S_110           3       /* 110 baud */
/*                      4       ;  134.5 baud */
#define S_150           5       /* 150 baud */
/*                      6       ;  200 baud */
#define S_300           7       /* 300 baud */
/*                      8       ;  600 baud */
#define S_1200          9       /* 1200 baud */
/*                      10      ;  1800 baud */
```

```
#define S_2400          11      /* 2400 baud */
#define S_4800          12      /* 4800 baud */
#define S_9600          13      /* 9600 baud */
/*                      14      ;  External A */
/*                      15      ;  External B */
#define S_19200         16      /*19200 baud */
#define S_CTSWAIT       125     /* wait for clear to send */
#define S_NOCHG         126     /* no change of baudrate */
#define S_UNINIT        127     /* uninitialized baudrate */
#define Sfl_AUTO        0x80    /* (bit 7): input CRs from keyboard to set
                                   baud */


/* contents of the d-register & e-register for MD_MODE1 calls
   stored in md_model (bit numbers) */

#define TANDEM          0x1
#define XTAB            0x2     /* expand TABs */
#define LCASE           0x4     /* convert alphabetics to lower case */
#define ECHO            0x8     /* echo input */
#define CRDEVICE        0x10    /* on input, map CR into NL */
                                /* on output, echo LF or CR as CRLF */
#define RAW             0x20    /* on input, return after each
                                   character */
                                /* and treat ^C, ^S, ^Q as regular
                                   input */
#define ODD             0x40    /* parity function bits */
#define EVEN            0x80

/* contents of the d-register & e-register for MD_MODED calls */
/* stored in md_moded */

#define NLDELAY         0x03    /* (pairs of bits) */
#define TABDELAY        0x0C
#define CRDELAY         0x30
#define FFDELAY         0x40    /* (single bits) */
#define BSDELAY         0x80

/* contents of the d-register & e-register for MD_MODE2 calls */
/* stored in md_mode2 (bit numbers) */

#define PAUSE           0x01    /* wait for CNTRL-Q after a page
                                           is output */
#define NOTIMMECHO      0x02    /* do not echo characters
                                           typed-ahead */
#define NOECNL          0x04    /* do not echo NLs */
#define SGENABLE        0x08    /* user-specifiable key signal enable */
#define ABENABLE        0x10    /* CNTRL-C key signal enable */
#define XFF             0x20    /* expand FFs */
#define WRAP            0x40    /* wrap-around if page width is exceeded */
#define SIGALLC         0x80    /* send siguser signal for each key pushed */

/* contents of the d-register & e-register for MD_MODE3 calls */
/* stored in md_mode3 (bit numbers) */
```

```
#define ESCRETN         0x01    /* ESC causes input line to be
                                   returned */
#define FNKEYS          0x02    /* enable response to 3102 function keys */
#define HUPENAB         0x04    /* hang up modem when device finally closed
#define SIGHUPALL       0x08    /* send sighangup signals to all processes
                                   which use this tty if modem hangs u
#define CBREAK          0x10    /* on input, return after each character, n
                                   erase, linekill, or eof characters
#define BINARY          0x20    /* on input, return after each character, n
                                   erase, linekill, or eof characters,
                                   no output pause or output width
                                   truncation, treat x-off, x-on as
                                   regular input, no tandem mode (ie,
                                   no input buff ctl), no abort signal
                                   (^c), no user signal, no changing o
                                   checking parity bit, no delays afte
                                   control chars as nls, no echoing, n
                                   character transformations, no
                                   function key decoding.  */
#define DISCARD         0x80    /* discard the device when it is no
                                   longer open */

/* bits of the d-register for MD_STATUS calls */

#define INOTEMPTY       0x1     /* there is a character in the input
                                   buffer */
                                /* (but if not RAW mode, it won't be
                                   accessible */
                                /*  until a whole line is entered) */

/* ------------------------------------------------------------------ */

/*                      Old names                          */

#define md_ibaud        MD_ISPEED
#define md_obaud        MD_OSPEED
#define md_mode1        MD_MODE1
#define md_mode2        MD_MODE2
#define md_mode3        MD_MODE3
/*  #define md_erase    MD_ERASE        */
/*  #define md_dlecho   MD_DELECHO      */
#define md_kill MD_LKILL
/*  #define md_signal   MD_USIGNAL      */
/*  #define md_length   MD_LENGTH       */
/*  #define md_width    MD_WIDTH        */
#define md_bmargin      MD_BMARGIN

#define md_status       MD_STATUS

/*  #define b_hangup    S_HANGUP        */
/*  #define b_110       S_110           */
/*  #define b_150       S_150           */
```

```
/*  #define b_300        S_300          */
/*  #define b_1200       S_1200         */
/*  #define b_2400       S_2400         */
/*  #define b_4800       S_4800         */
/*  #define b_9600       S_9600         */
/*  #define b_19200      S_19200        */
/*  #define b_auto       Sfl_AUTO       */
/*  #define b_exta       14             */
/*  #define b_extb       15             */

/*  #define mdl_tab      XTAB           */
/*  #define mdl_lcase    LCASE          */
/*  #define mdl_echo     ECHO           */
/*  #define mdl_cr_nl    CRDEVICE       */
/*  #define mdl_raw      RAW            */
/*  #define mdl_odd      ODD            */
/*  #define mdl_even     EVEN           */

/*  #define md2_pause    PAUSE          */
/*  #define md2_later    NOTIMMECHO     */
/*  #define md2_noecnl   NOECNL         */
/*  #define md2_sgenable  SGENABLE      */
/*  #define md2_abenable  ABENABLE      */
/*  #define md2_ff       XFF            */
/*  #define md2_wrap     WRAP           */
/*  #define md2_esccr    ESCRETN        */

/*  #define st_charrdy   INOTEMPTY      */
/*  #define hangup       HUPENAB        */
/*  #define huptty       SIGHUPALL      */

#control list
```

Structs.h

```
#control list

/* Structs.h:        Cromemco C I/O header file

   Copyright (c) 1982 by Cromemco, Inc., All Rights Reserved

   This file contains declarations of all values which are used to
   reference the super block and inodes in the Cromix Operating System.

  super block definitions
*/

#define frbcount     80                /* free block list size */
#define fricount     80                /* free inode list size */
#define frbsize      frbcount*4+2      /* free list size in bytes */
#define frisize      fricount*2+2      /* free list size in bytes */

struct  sblock {
        char         version[2];       /* version number */
        char         cromix[6];        /*  'cromix' */
        unsigned     istart;           /* first inode block */
        unsigned     isize;            /* number of inodes */
        long         fsize;            /* max block number */
        char         time[6];          /* last modified time */
        char         dummy0[6];        /* Unused space */
        unsigned     nfree;            /* free block count */
        long         free[frbcount];   /* free list address */
        unsigned     ilist;            /* i-list address */
        unsigned     inode[fricount]; /* free inodes */
};
#define ninode  ilist  /* free inode count */


/*      inode buffer definitions        */

struct inode
{
        unsigned     owner;            /* file owner's user id */
        unsigned     group;            /* file owner's group id */
        char         aowner;           /* owner access */
        char         agroup;           /* group access */
        char         aother;           /* other access */
        char         stat;             /* file status */
        char         nlinks;           /* number of links to inode*/
        char         dummy3;           /* defs 1 */
        long         size;             /* file total size (in bytes) */
        unsigned     inode;            /* this inode number */
        unsigned     parent;           /* parent inode number
                                          (for directories only)*/
        unsigned     dcount;           /* number entries in a directory */
        long         usage;            /* num blocks actually used in file*/
```

```
        char            tcreate[6];     /* time created */
        char            tmodify[6];     /* time last modified */
        char            taccess[6];     /* time last accessed */
        char            tdumped[6];     /* time last dumped (backed up) */
        long            index[20];      /* block pounsigneders */
};

#define inosize sizeof(inode)           /* total inode size in bytes */
#define begin owner                     /* beginning of inode on disk */
#define sdevn dcount                    /* special device major & minor
                                           numbers */

#define inocount  20                    /* size of inode table */

#define is_type         0x80            /* file type mask (stat) */
#define is_ordin        0x01            /* ordinary file */
#define is_direct       0x02            /* directory file */
#define is_char         0x04            /* character device */
#define is_block        0x08            /* block device */
#define is_pipe         0x10            /* pipe file */
#define is_alloc        0x80            /* inode allocated (bit in stat) */
#define if_lock         0x01            /* inode locked (in use by a process)
#define if_want         0x02            /* inode wanted by another process */
#define if_modf         0x04            /* inode has to be written out */
#define if_modt         0x08            /* update time modified */
#define if_acct         0x10            /* update time accessed */
#define ac_read         0x01            /* read access bit */
#define ac_exec         0x02            /* execute access bit */
#define ac_writ         0x04            /* write access bit */
#define ac_apnd         0x08            /* append access bit */

/*  directory format definitions */

struct dir
{
        char            name[24];       /* name of entry */
        char            reserved[4];    /* reserved */
        unsigned        stat;           /* status & flags */
        unsigned        inum;           /* inode number of file */
};
#define dirsize 32                      /* directory entry size (32 bytes) */
#define namsiz  24                      /* Max file name size */

#define ds_alloc 7                      /* entry allocated bit */
#control list
```

VERSION NUMBER SUMMARY

File	Version
cp0.bin	05.10
cp1.bin	05.10
cp2.bin	05.10
asmb.com	03.08
debug.com	00.17
lib.com	none
link.com	03.44